组件注册:

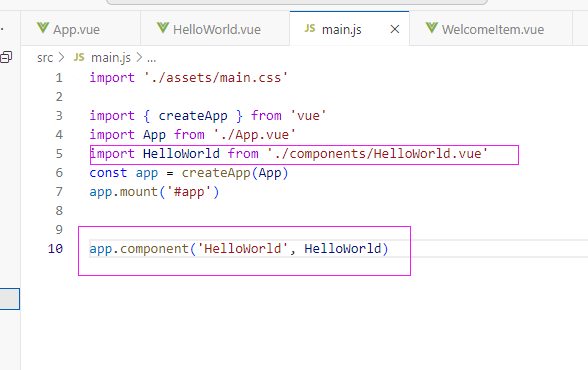
分为全局和局部注册：

全局注册是在main.js里

写一个组件HelloWorld：



在main.js中加入：



就把676767展示在页面：

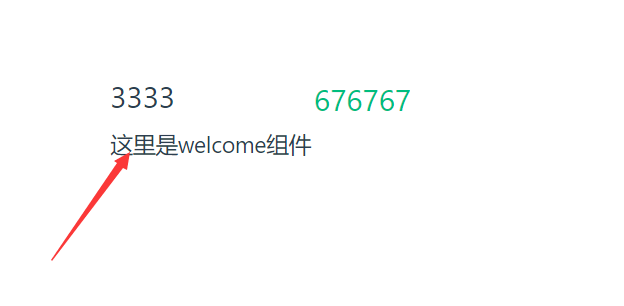


局部注册：

在app.vue中局部注册Welcome组件：



页面显示：



Props : 父组件向子组件传值（props）、子组件向父组件传值（emit）defineProps() 宏:defineProps是编译器宏，无需引入），（defineProps作用是****拿父组件传过来的值）****

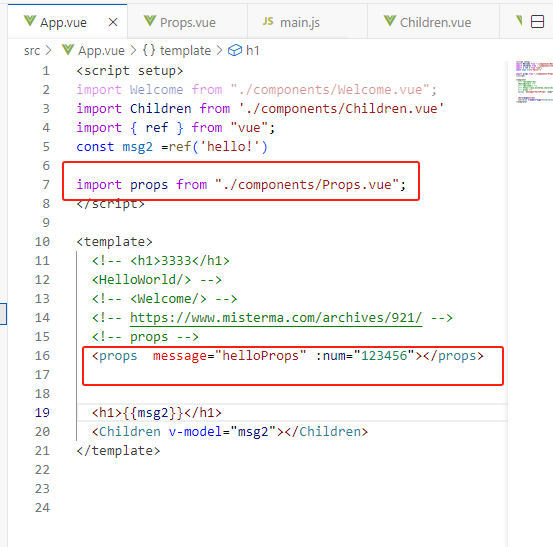
https://juejin.cn/post/7242676549736251450?from=search-suggest

单文件组件<script setup>在 <script setup> 中必须使用 defineProps API 来声明 props就是说setup中，我们使用defineProps来定义父组件传递的props：

写一个子组件：



父组件：



父组件的字符串helloProps和数字123456就被传给了子组件对应的类型的props变量中，定义后props值可直接在模板中使用或赋值给data对象。传数值类型或者复杂类型时需要加‘：’传递。由defineProps定义出的props对象，是一个proxy对象，所有特性和reactive基本相同，只不过由defineProps定义出的props对象的**值是只读**的，还有在模板上可以单独属性直接使用。

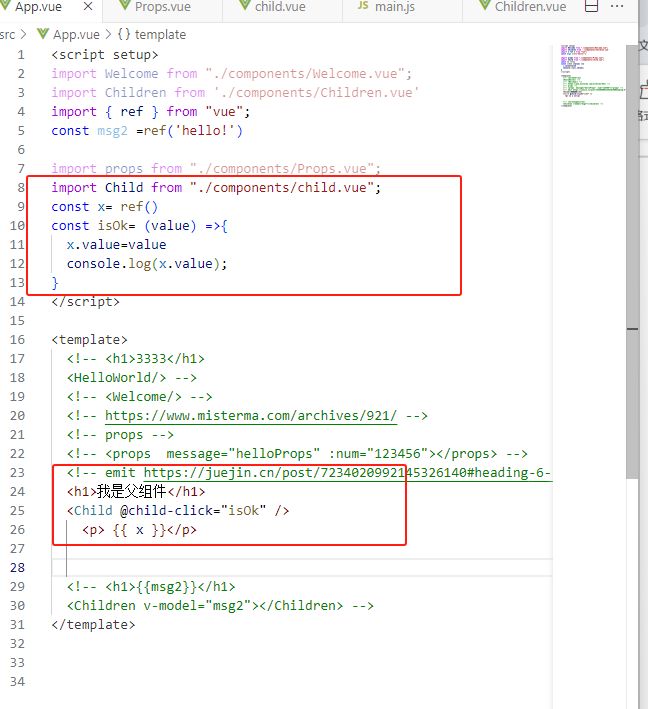
需要注意的是：defineProps只能在setup中使用，且只能在setup的顶层使用，不能在局部作用域使用。defineProps声明的props的值是只读的，不能修改，当传过来的是原始值时不能修改，如果是对象类型时，不能修改对象类型的引用地址。

既然有父传子那就有子传父.可以使用Emit使用，

先声明一个子组件：



父组件里：

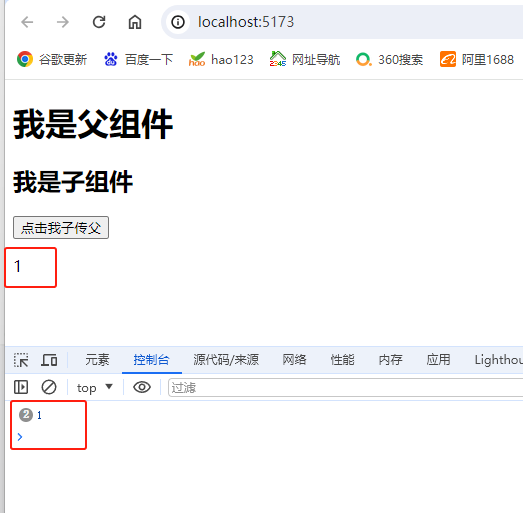


页面显示：

每点击之前：



点击之后：



这个1 是子组件isOk方法的值1



子组件里面点击按钮，触发了isOk方法后，该方法里面又通过emit触发了child-click方法，子组件里的emit可以触发到父组件里的方法，当子组件触发自定义事件（child-click）时，父组件可以使用 v-on指令(或简写语法"@")来监听该事件，并在事件触发时调用指定的方法。

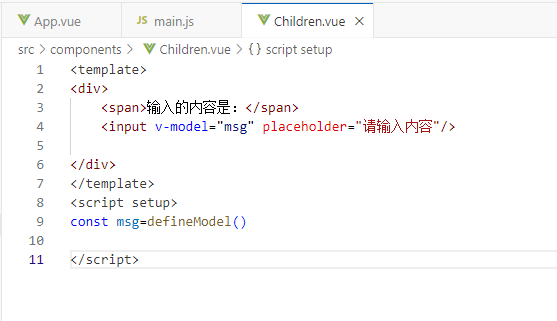
事件：@符号用来监听事件：

v-model：组件之间用于双向绑定：

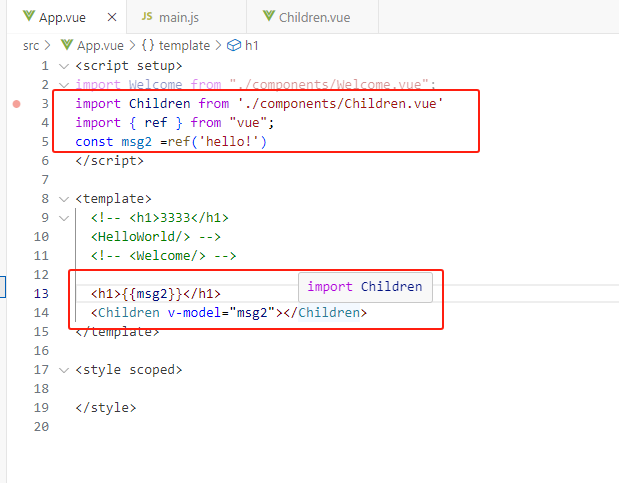
使用宏的话就可以起到在父组件和当前变量之间的双向绑定的作用：子组件的value 可以和v-model的值同步，子组件发生变化时，触发父组件绑定的值一起更新.

基本用法：

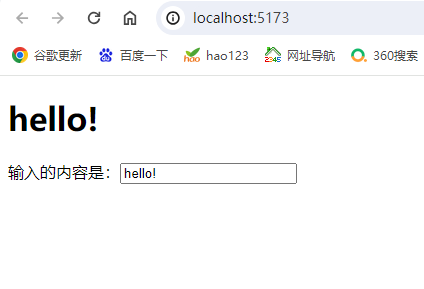
写一个子组件：有一个输入框：



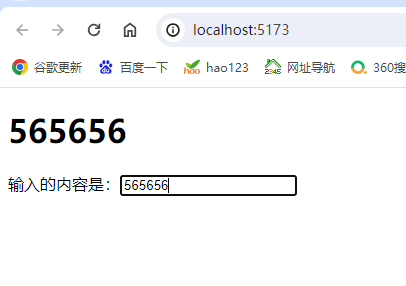
在app.vue文件中写入：



先看页面显示：



对输入框的文字进行修改：



可以看到输入框的值变化的时候最上面的一行的值也对应发生变化。

主页面定义了msg2的ref数据，把msg2绑定到children组件之上，children组件定义了一个输入框，输入框msg是宏，当子组件的输入框发生变化的时候主页面的msg2的也对应发生变化。两个组件之间的变化是绑定的，实现表单元素与组件数据之间的双向绑定。